

Generalized Modal Satisfiability [★]

Edith Hemaspaandra, Henning Schnoor, and Ilka Schnoor

Department of Computer Science, Rochester Institute of Technology, Rochester, NY
14623, U.S.A. {eh,hs,is}@cs.rit.edu

Abstract. It is well known that modal satisfiability is PSPACE-complete [Lad77]. However, the complexity may decrease if we restrict the set of propositional operators used. Note that there exist an infinite number of propositional operators, since a propositional operator is simply a Boolean function. We completely classify the complexity of modal satisfiability for every finite set of propositional operators, i.e., in contrast to previous work, we classify an infinite number of problems. We show that, depending on the set of propositional operators, modal satisfiability is PSPACE-complete, coNP-complete, or in P. We obtain this trichotomy not only for modal formulas, but also for their more succinct representation using modal circuits. We consider both the uni-modal and the multi-modal case, and study the dual problem of validity as well.

Keywords: computational complexity, modal logic

1 Introduction

Modal logics are valuable tools in computer science, since they are often a good compromise between expressiveness and decidability. Standard applications of modal logics are in artificial intelligence [Moo79,MSHI78], and cryptographic and other protocols [FHJ02,CDF03,HMT88,LR86]. More recent applications include a new modal language called Versatile Event Logic [BG04], and the usage to characterize the relationship among belief, information acquisition, and trust [Lia03].

Applications of modal logic for solving practical problems obviously require a study of the computational complexity of various aspects of modal logics. A central computational problem related with any logic is the satisfiability problem, that is to decide whether a given formula has a model. The first complexity results for the modal satisfiability problem were achieved by Ladner [Lad77]. He showed that the basic modal satisfiability problem is PSPACE-complete. There is a rich literature on the complexity of variants of the modal satisfiability problem, important works include the paper by Halpern and Moses [HM92]

[★] Supported in part by the DAAD Postdoc Program, by grants NSF-CCR-0311021, NSF-IIS-0713061, and DFG VO 630/5-1, and by a Friedrich Wilhelm Bessel Research Award. Work done in part while the second and third authors worked at the Leibniz Universität Hannover. An earlier version of some of the results appeared as [BHSS06].

on multi-modal logics. Recently, PSPACE-algorithms for a wide class of modal logics were presented by Schröder and Pattinson [SP06].

For modal logics to be used in practice, a lower complexity of the satisfiability problem than the aforementioned PSPACE-hardness is desirable. It turns out that for many applications, the full power of modal logic is not necessary. There are various ways of defining restrictions of modal logics which potentially lead to a computationally easier version of the satisfiability problem that have been studied: Variations of modal logics are achieved by restricting the class of considered models, e.g., instead of allowing arbitrary graphs, classical examples of logics only allow reflexive, transitive, or symmetric graphs as models. Many complexity results for logics defined in this way have been achieved: Initial results for many important classes are present in the above-mentioned work by Ladner [Lad77]. Recently, Hemaspaandra and Schnoor considered a uniform generalization of many of these examples [HS08]. It should be noted that such restrictions do not necessarily decrease the complexity; for many common restrictions, the complexity remains the same [Lad77, HM92] and it is even possible that the complexity increases. In [Hem96], Hemaspaandra showed that the complexity of the global satisfiability problem increases from EXPTIME-complete to undecidable by restricting the graphs to those in which every node has at least two successors and at most three 2-step successors.

Another way of restricting modal logics is to change the syntax rather than the semantics, i.e., restrict the structure of the considered modal formulas. Syntactical restrictions are known to naturally reduce the complexity of many decision problems in logic. In propositional logic, well-known examples are the satisfiability problems for Horn formulas, 2CNF formulas, or formulas describing monotone functions: All of these can be solved in polynomial time, while the general propositional satisfiability problem is NP-complete. Syntactical restrictions have been considered in the context of modal logics before: Halpern showed that the complexity of the modal satisfiability problem decreases to linear time when restricting the number of variables and nesting degree of modal operators [Hal95]. Restricted modal languages where only a subset of the relevant modal operators are allowed have been studied in the context of linear temporal logic (see, e.g., [SC85]). Some description logics can be viewed as modal logic with a restriction on the propositional operators that are allowed. For the complexity of description logics, see, e.g., [SS91, DHL⁺92, DLNN97]. For the complexity of modal logic with other restrictions on the set of operators, see [Hem01].

The approach we take in the present paper is to generalize the occurring propositional operators in the formulas. Instead of the operators \wedge, \vee and negation, we allow the appearing operators to represent arbitrary Boolean functions. In particular, there are an infinite number of Boolean operators. We completely classify the complexity of modal satisfiability for every finite set of propositional operators. The restriction on the propositional operators leads to a classification following the structure of Post's Lattice [Pos41], a tool that has been applied in similar contexts before: For propositional logic, Lewis showed that the satisfiability problem is dichotomic: Depending on the set of operators, propositional satis-

fiability is either NP-complete or solvable in polynomial time [Lew79]. For modal satisfiability, we achieve a trichotomy: For the modal logic K, the satisfiability problem is PSPACE-complete, coNP-complete, or in P. We also achieve a full classification for the logic KD (in this case, we show a PSPACE/P-dichotomy), and almost complete classifications for the logics T, S4, and S5.

When considering sets of operations which do not include negation, the complexity for the cases where one modal operator is allowed sometimes differs from the case where we allow both operator \Diamond and its dual operator \Box . With only one of these, modal satisfiability is PSPACE-complete exactly in those cases in which propositional satisfiability is NP-complete. When we allow both modal operators, the jump to PSPACE-completeness happens earlier, i.e., with a set of operations with less expressive power.

We consider several generalizations of the problems outlined above. In particular, we introduce *modal circuits* as a succinct way of representing modal formulas. We show that this does not give us a significantly different complexity than the formula case. We also consider multi-modal logics, in which several independent modal operators are introduced.

In addition to the satisfiability problem, we also study the validity problem, where we do not ask whether a formula is satisfiable, but whether it is true in every possible model. Since our restricted modal languages do not always include negation, the complexity of this problem turns out to be different from, but related to, the complexity of the satisfiability problem.

An interesting case in our classifications is the case where we only allow the propositional exclusive-or and constants as propositional operators. For purely propositional logics, it is very easy to see that satisfiability for these formulas (essentially linear equations over GF(2)) can be decided in polynomial time. In the case of modal logics, an analogous result holds, but the proof requires significantly more work. As in the propositional case, it yields an optimal solution to the *minimization problem* as well: Given a modal formula or modal circuit using only these propositional operators, we can efficiently compute an equivalent formula or circuit of minimal size.

The structure of the paper is as follows: In Section 2, we introduce the necessary definitions, recall results from the literature, and prove some basic facts about our problems. Section 3 contains our main results: The complete classification of the complexity of the modal satisfiability problem for every possible set of Boolean operators. In Section 4 we prove a relationship between satisfiability and validity implying a full classification of this problem as well. We conclude in Section 5 with some open questions for future research.

2 Preliminaries

2.1 Modal Logic

Modal logic is an extension of classical propositional logic that talks about “possible worlds.” We first introduce the usual uni-modal logic, and then generalize it

to the multi-modal case. Uni-modal logics enrich the vocabulary of propositional logic with an additional unary modal operator \Diamond . A model for a given formula consists of a directed graph with propositional assignments. To be more precise, a *frame* consists of a set W of “worlds,” and a “successor” relation $R \subseteq W \times W$. For $(w, w') \in R$, we say w' is a *successor* of w . A *model* M consists of a frame (W, R) , a set X of propositional variables, and a function $\pi: X \rightarrow \mathcal{P}(W)$. The intuition is that for $x \in X$, $\pi(x)$ denotes the set of worlds in which the variable x is true. The operator \Box is the dual operator to \Diamond , $\Box\varphi$ is defined as $\neg\Diamond\neg\varphi$. Intuitively, $\Diamond\varphi$ means “there is a successor world in which φ holds,” and $\Box\varphi$ means “ φ holds in all successor worlds.” For a class \mathcal{F} of frames, we say a model M is an \mathcal{F} -model if the underlying frame is an element of \mathcal{F} .

In multi-modal logic, a finite number of these modal operators is considered, where each operator \Diamond_i corresponds to an individual successor relation R_i . For a modal logic with k modalities, a frame again consists of a set W of worlds, and successor relations $R_1, \dots, R_k \subseteq W \times W$. If $(w, w') \in R_i$, we say that w' is a *i-successor* of w . For a formula φ built over the variables X , propositional operators \wedge and \neg , and the unary modal operators $\Diamond_1, \dots, \Diamond_k$, we define what “ φ holds at world w ” means for a model M (or M, w *satisfies* φ) with assignment function π , written as $M, w \models \varphi$.

- If φ is a propositional variable x , then $M, w \models \varphi$ if and only if $w \in \pi(x)$,
- $M, w \models \varphi_1 \wedge \varphi_2$ if and only if $(M, w \models \varphi_1 \text{ and } M, w \models \varphi_2)$,
- $M, w \models \neg\varphi$ if and only if $M, w \not\models \varphi$,
- for $i \in \{1, \dots, k\}$, $M, w \models \Diamond_i\varphi$ if and only if there is a world $w' \in W$ such that $(w, w') \in R_i$ and $M, w' \models \varphi$.

Analogously to the unimodal case, the operator \Box_i is defined as $\Box_i\varphi = \neg\Diamond_i\neg\varphi$. For a class \mathcal{F} of frames, we say a formula φ is \mathcal{F} -satisfiable if there exists an \mathcal{F} -model $M = (W, R, \pi)$ and a world $w \in W$ such that $M, w \models \varphi$. For modal formulas φ and ψ , we write $\varphi \equiv_{\mathcal{F}} \psi$ if for every world in every \mathcal{F} -model, φ holds if and only if ψ holds. Note that a formula φ is \mathcal{F} -satisfiable iff $\varphi \not\equiv_{\mathcal{F}} 0$. Similarly, we say that φ is an \mathcal{F} -tautology if $\varphi \equiv_{\mathcal{F}} 1$, and finally φ is \mathcal{F} -constant if $\varphi \equiv_{\mathcal{F}} 0$ or $\varphi \equiv_{\mathcal{F}} 1$.

K	All frames
KD	Frames in which every world has a successor
K4	Transitive frames
S4	Frames that are reflexive and transitive
S5	Frames that are reflexive, transitive, and symmetric
T	Reflexive frames

Table 1. Classes of frames

We now define the classes of frames that are most commonly used in applications of modal logic. To see how these frames correspond to axioms and proof systems, see, for example, [BdRV01, Section 4.3]. Again, we first consider the uni-modal case and then present the natural generalizations to multi-modal logics. K is the class of all frames, KD is the class of frames in which every world has a successor, i.e., for all $w \in W$, there is a $w' \in W$ such that $(w, w') \in R$. T is the class of reflexive frames, $K4$ is the class of transitive frames, $S4$ is the class of frames that are both reflexive and transitive, and $S5$ is the class of reflexive, symmetric, and transitive frames. The *reflexive singleton* is the frame consisting of one world w , and the relation $\{(w, w)\}$. Note that all classes of frames \mathcal{F} described above contain the reflexive singleton. Similarly, the *irreflexive singleton* is the frame consisting of one world, and an empty successor relation.

For multi-modal logics, the generalizations are obvious: For a class of frames \mathcal{F} as previously defined, we say that the class \mathcal{F}_k contains those frames (W, R_1, \dots, R_k) , where $(W, R_i) \in \mathcal{F}$ for all $i \in \{1, \dots, k\}$. In particular, a multi-modal reflexive singleton consists of the set of worlds $W = \{w\}$ where each successor relation consists of the pair (w, w) , and the multi-modal irreflexive singleton consists of the same set of worlds where all of the successor relations are empty. If the number k of modal operators is clear from the context, we often simply write \mathcal{F} instead of \mathcal{F}_k , speak about the reflexive singleton, etc.

2.2 Generalized Formulas and Circuits

We now consider a more general notion of modal formulas, whose propositional analog has been studied extensively. We generalize the notion of a modal formula in two ways: First, instead of allowing the usual propositional operators \wedge, \vee , and \neg , we allow arbitrary Boolean functions. Second, we study circuits as succinct representations of formulas. Intuitively, a circuit is a generalization of a formula in the same way as a directed acyclic graph is a generalization of a tree, since formulas directly correspond to tree-like circuits. To be more precise, for a finite set B of Boolean functions, a *modal B -circuit* is a generalization of a propositional Boolean circuit (see e.g., [Vol99] for an introduction to Boolean circuits) with gates for functions from B and additional gates representing the modal operators \Diamond_i or \Box_i . Boolean circuits are a standard way to succinctly represent Boolean functions. Formally, we define the following (recall that X is the set of propositional variables):

Definition 2.1. *Let B be a finite set of Boolean functions, and let $M \subseteq \{\Box, \Diamond\}$. A circuit in $\text{MCIRC}_M^k(B)$ is a tuple $C = (V, E, \alpha, \beta, \text{out})$ where (V, E) is a finite directed acyclic graph, $\alpha: E \rightarrow \mathbb{N}$ is an injective function, $\beta: V \rightarrow B \cup \{\Box_1, \dots, \Box_k, \Diamond_1, \dots, \Diamond_k\} \cup X$ is a function, and $\text{out} \in V$, such that*

- *If $v \in V$ has in-degree 0, then $\beta(v) \in X$ or $\beta(v)$ is a 0-ary function (a constant) from B .*
- *If $v \in V$ has in-degree 1, then $\beta(v)$ is a 1-ary function from B or, for some $i \in \{1, \dots, k\}$, one of the operators \Box_i (if $\Box \in M$) or \Diamond_i (if $\Diamond \in M$).*

- If $v \in V$ has in-degree $d > 1$, then $\beta(v)$ is a d -ary function from B .

By definition, $\text{MCIRC}_M^k(B)$ contains the modal circuits that use the following as operators: functions from B , the modal operators $\Diamond_1, \dots, \Diamond_k$ if $\Diamond \in M$, and \Box_1, \dots, \Box_k if $\Box \in M$. Nodes $v \in V$ are called *gates* of C , $\beta(v)$ is the *gate-type* of v . The node *out* is the *output-gate* of C . The function α is needed to define the order of arguments for non-commutative functions. The *size* of a modal circuit C is the number of gates: $|C| := |V|$.

In addition to circuits, we also study the special case of modal formulas. A *modal B-formula* is a modal B -circuit where each gate has out-degree ≤ 1 . This corresponds to the intuitive idea of a formula: Such a circuit can be written down as a formula, e.g., in prefix notation, without growing significantly in size. Semantically we interpret a circuit as a succinct representation of its formula expansion. For a modal B -circuit C , the *modal depth* of C , $md(C)$, is the maximal number of gates representing modal operators on a directed path in the graph. If there are no modal gates (i.e., gates $v \in C$ such that $\beta(v) \in \{\Box_i, \Diamond_i\}$ for any i) then φ_C is a *propositional Boolean formula* and C is a *propositional Boolean circuit*.

In order to define the semantics of the circuits defined above, we relate them to formulas in the following natural way: The circuit C represents the modal formula φ_C that is inductively defined by a modal B -formula φ_v for every gate v in C :

- Definition 2.2.** – If $v \in V$ has in-degree 0, then $\varphi_v := \beta(v)$.
- Let $v \in V$ have in-degree $l > 0$, and let v_1, \dots, v_l be the predecessor gates of v such that $\alpha((v_1, v)) < \dots < \alpha((v_l, v))$. Then let $\varphi_v := \beta(v)(\varphi_{v_1}, \dots, \varphi_{v_l})$.
 - Finally, we define φ_C as φ_{out} . We call φ_C the *formula expansion* of C .

Since every Boolean function can be expressed using only conjunction and negation, the semantics for circuits allowing arbitrary Boolean functions is immediate. It is obvious from the definition that for every modal circuit, there is an equivalent formula. Therefore, considering circuits instead of formulas does not increase the expressive power, but circuits are a succinct representation of formulas (there are circuits representing formulas where the size of the formula is exponential in the size of the circuit).

2.3 Problem Definitions

We now define the various modal satisfiability problems we are interested in. As usual in computational complexity, we define the problems as the sets of their yes-instances.

Definition 2.3. Let B be a finite set of Boolean functions, \mathcal{F} a class of frames, $k \geq 0$, and $M \subseteq \{\Diamond, \Box\}$. Then

- $\text{MFORM}_M^k(B)$ is the set of formula expansions of circuits in $\text{MCIRC}_M^k(B)$, i.e., the set of modal formulas using operators from B , and modalities \Box_1, \dots, \Box_k (if $\Box \in M$) and $\Diamond_1, \dots, \Diamond_k$ (if $\Diamond \in M$).

- $\mathcal{F}\text{-FSAT}_M^k(B)$ is the set of \mathcal{F}_k -satisfiable formulas from $\text{MFORM}_M^k(B)$.
- $\mathcal{F}\text{-CSAT}_M^k(B)$ is the set of \mathcal{F}_k -satisfiable circuits from $\text{MCIRC}_M^k(B)$.
- $\mathcal{F}\text{-FTAUT}_M^k(B)$ is the set of \mathcal{F}_k -tautologies in $\text{MFORM}_M^k(B)$,
- $\mathcal{F}\text{-CTAUT}_M^k(B)$ is the set of \mathcal{F}_k -tautologies in $\text{MCIRC}_M^k(B)$.

For readability, we often leave out the set brackets and write, for example, $\text{K-FSAT}_{\square}^1(\oplus, 1)$ instead of $\text{K-FSAT}_{\{\square\}}^1(\{\oplus, 1\})$. In addition to specifying whether \diamond and \square are allowed “globally,” we could also allow our model to specify for each $i \in \{1, \dots, k\}$ whether \diamond_i and \square_i are allowed to appear in the circuits. However, our hardness results usually require only a single one of these operators to be present (and upper complexity bounds obviously transfer to the restricted setting). Therefore, the definition we gave captures the significant variations of the problems we study.

From the definitions, the following is immediate, which we will often use without reference. It is obvious that analogous results hold for the tautology problem as well. Due to this proposition, it is clear that it suffices to state lower complexity bounds for the problems involving formulas, and upper bounds for the problems involving circuits.

Proposition 2.4. *Let $B_1 \subseteq B_2$ be finite sets of Boolean functions, let \mathcal{F} be a class of frames, let $k_1 \leq k_2$, and let $M_1 \subseteq M_2 \subseteq \{\square, \diamond\}$. Then the following hold:*

- $\mathcal{F}\text{-FSAT}_{M_1}^{k_1}(B_1) \leq_m^{\log} \mathcal{F}\text{-FSAT}_{M_2}^{k_2}(B_2)$,
- $\mathcal{F}\text{-FSAT}_{M_1}^{k_1}(B_1) \leq_m^{\log} \mathcal{F}\text{-CSAT}_{M_2}^{k_2}(B_2)$,
- $\mathcal{F}\text{-CSAT}_{M_1}^{k_1}(B_1) \leq_m^{\log} \mathcal{F}\text{-CSAT}_{M_2}^{k_2}(B_2)$.

Initial complexity results can be found in the literature; we state them in our notation:

Theorem 2.5 ([HM92],[Lad77]).

1. $\text{S5-FSAT}_{\square}^1(\wedge, \neg)$ is NP-complete.
2. Let $\mathcal{F} \in \{\text{K}, \text{KD}, \text{K4}, \text{T}, \text{S4}\}$. Then $\mathcal{F}\text{-FSAT}_{\square}^1(\wedge, \neg)$ is PSPACE-complete.
3. Let $\mathcal{F} \in \{\text{K}, \text{KD}, \text{K4}, \text{T}, \text{S4}, \text{S5}\}$, and let $k \geq 2$. Then $\mathcal{F}\text{-FSAT}_{\square}^k(\wedge, \neg)$ is PSPACE-complete.

In [Hem01], Hemaspaandra examined the complexity of $\text{K-FSAT}_M^1(B)$ for all $M \subseteq \{\square, \diamond\}$ and $B \subseteq \{\wedge, \vee, \neg, 0, 1\}$. In this paper, we generalize this result in several ways: We classify the complexity of modal satisfiability for all finite sets of Boolean functions (in particular, we determine the complexity of an infinite number of problems), and we consider multi-modal logic as well. Further, we also consider the case of circuits instead of formulas, and study different frame classes. Finally, we also consider the validity problem.

2.4 Clones and Post's Lattice

The notion of clones is very helpful to bring structure to this infinite set of problems. We introduce the necessary definitions, and some important properties of Boolean functions. An n -ary function f is a *projection function* if there is some i such that for all $\alpha_1, \dots, \alpha_n \in \{0, 1\}$, $f(\alpha_1, \dots, \alpha_n) = \alpha_i$. A set B of Boolean functions is called a *clone* if it is closed under *superposition*, that is, B contains all projection functions and is closed under permutation of variables, identification of variables, and arbitrary composition. It is easy to see that the set of clones forms a lattice. Post determined the complete set of clones, as well as their inclusion structure [Pos41]. A graphical presentation of the lattice of clones, also known as Post's Lattice, can be found in Figure 1. For a set B of Boolean functions, let $[B]$ be the smallest clone containing B .

We briefly define the clones that arise in our complexity classification. The smallest clone contains only projections and is named I_2 . Further, $I_1 = [\{1\}]$. The largest clone $BF = [\{\wedge, \neg\}]$ is the set of all Boolean functions. The set of all monotone functions forms a clone denoted by $M = [\{\vee, \wedge, 0, 1\}]$. D consists of all *self-dual* functions, i.e., $f \in D$ if and only if $f(x_1, \dots, x_n) = \neg f(\bar{x}_1, \dots, \bar{x}_n)$. $L = [\{\oplus, 1\}]$ is the set of all linear Boolean functions (where \oplus is the Boolean exclusive or). The clone of all Boolean functions that can be written using only disjunction and constants is called $V = [\{\vee, 1, 0\}]$; further, $V_0 = [\{\vee, 0\}]$ and $V_2 = [\{\vee\}]$. Similarly, the clone $E = [\{\wedge, 0, 1\}]$ contains the Boolean functions that can be written as conjunctions of variables and constants; $E_0 = [\{\wedge, 0\}]$ and $E_2 = [\{\wedge\}]$. R_1 is built from all *1-reproducing* functions, i.e., all functions f satisfying $f(1, \dots, 1) = 1$. The clone $N = [\{\neg, 1\}]$ consists of the projections, their negations, and all constant Boolean functions. $S_1 = [\{x \wedge \bar{y}\}]$ and $S_{11} = S_1 \cap M$.

BF	All Boolean functions
S_1	$[x \wedge \bar{y}]$
M	Monotone functions
S_{11}	$M \cap S_1$
R_1	f with $f(1, \dots, 1) = 1$
D	Self-dual functions
L	Linear functions
V	Multi-ary OR and constants 0, 1
V_0	Multi-ary OR and constant 0
V_2	Multi-ary OR
E	Multi-ary AND and constants 0, 1
E_0	Multi-ary AND and constant 0
E_2	Multi-ary AND
N	Negation, identity, and constants
I	Identity and constants

9

If we interpret Boolean formulas as Boolean functions, then $[B]$ consists of all propositional formulas that are equivalent to a formula built with variables and operators from B . Therefore, this framework can be used to investigate problems related to Boolean formulas depending on which connectives are allowed. Several problems have been studied in this context: Lewis proved that the satisfiability problem for Boolean formulas with connectives from B is NP-complete if $S_1 \subseteq [B]$ and in P otherwise [Lew79]. Another example is the classification of the equivalence problem given by Reith: Deciding whether two formulas with connectives from B are equivalent is in LOGSPACE if $[B] \subseteq V$ or $[B] \subseteq E$ or $[B] \subseteq L$, and coNP-complete in all other cases [Rei01]. Dichotomy results for counting the solutions of formulas [RW05], finding the minimal solutions of formulas [RV00], and learnability of Boolean formulas and circuits [Dal00] were achieved as well. After presenting our results in [BHSS06], analogous classifications have been achieved by Bauland et al. in the context of temporal logics [BSS⁺07, BMS⁺07].

Post's Lattice has also been a helpful tool in the constraint satisfaction context. It can be used to obtain a very easy proof of Schaefer's Theorem [Sch78] and related complexity classifications. This is surprising, because constraint satisfaction problems are not related to Post's Lattice by definition, but clones appear indirectly through a Galois connection [JCG97]. For more information about the use of Post's Lattice in complexity classifications of propositional logic, see, for example, [BCRV03, BCRV04]. Finally, the notion of clones is not restricted to the Boolean case, but has been studied for arbitrary domains. The monograph [Lau06] is an excellent survey of clone theory.

The structure given by Post's Lattice enables us to compare the complexity of our circuit-related problems for the cases in which the corresponding clones are comparable. For circuits, we get a stronger result than Proposition 2.4: The complexity of our problems does not depend on the actual set B of Boolean functions, but just on the clone $[B]$ generated by it. Again, an analogous result holds for the tautology problem.

Lemma 2.6. *Let B_1, B_2 be finite sets of Boolean functions, \mathcal{F} a class of frames, $k \geq 1$, and $M \subseteq \{\diamond, \square\}$. If $B_1 \subseteq [B_2]$, then $\mathcal{F}\text{-CSAT}_M^k(B_1) \leq_m^{\log} \mathcal{F}\text{-CSAT}_M^k(B_2)$.*

Proof. This reduction is achieved by replacing every occurring gate representing a function from B_1 with the appropriate B_2 -circuit computing the same function. The resulting circuit obviously is \mathcal{F} -equivalent to the original circuit. \square

It is worth noting that an analogous result for formulas cannot be obtained in such an easy way, as the following example illustrates: Consider the sets $B_1 = \{\oplus\}$ and $B_2 = \{\wedge, \vee, \neg\}$ of Boolean functions. Since every Boolean function can be represented using only AND, OR, and negation gates, it is obvious that $B_1 \subseteq [B_2]$ holds. However, a reduction from $\text{K-FSAT}_\emptyset^0(B_1)$ to $\text{K-FSAT}_\emptyset^0(B_2)$ cannot be achieved in a straightforward manner, as a formula transformation analogous to the proof of Lemma 2.6 would replace a subformula $\varphi_1 \oplus \varphi_2$ with the formula $(\varphi_1 \wedge \neg \varphi_2) \vee (\neg \varphi_1 \wedge \varphi_2)$, and repeated application of this transformation

leads to exponential size for nested formulas. However, we will see that in the cases arising in this paper, the complexity of a problem $\mathcal{F}\text{-FSAT}_M^k(B)$ also only depends on the clone generated by B .

3 The Satisfiability Problem

Our main results are the classification theorems which we will present now. A graphical presentation of these results can be found in Figures 2 and 3. For the most general problem of K-satisfiability, we get the following trichotomy:

Theorem 3.1. *Let B be a finite set of Boolean functions, $k \geq 1$, and $\emptyset \neq M \subseteq \{\Box, \Diamond\}$. Then the following holds:*

- If $B \subseteq R_1, D, V$, or L , then $\text{K-FSAT}_M^k(B), \text{K-CSAT}_M^k(B) \in \text{P}$ (Corollary 3.15, Theorem 3.18, Theorem 3.19).
- If $E_0 \subseteq [B] \subseteq E$, then $\text{K-FSAT}_M^k(B), \text{K-CSAT}_M^k(B) \in \text{P}$ if $|M| \leq 1$, and are coNP-complete otherwise (Section 3.3, Theorem 3.23).
- if $S_{11} \subseteq [B] \subseteq M$, and $\text{K-FSAT}_M^k(B)$ and $\text{K-CSAT}_M^k(B)$ are PSPACE-complete if $M = \{\Box, \Diamond\}$, and in P otherwise (Corollary 3.11, Theorem 3.23).
- Otherwise, $S_1 \subseteq [B]$ and $\text{K-FSAT}_M^k(B)$ and $\text{K-CSAT}_M^k(B) \in \text{P}$ are PSPACE-complete (Corollary 3.11).

For the logic KD, we get the following complete classification:

Theorem 3.2. *Let B be a finite set of Boolean functions, $k \geq 1$, and $\emptyset \neq M \subseteq \{\Box, \Diamond\}$. Then the following holds:*

- If $B \subseteq R_1, D, M$, or L , then $\text{KD-FSAT}_M^k(B), \text{KD-CSAT}_M^k(B) \in \text{P}$ (Corollary 3.15, Theorem 3.16, Theorem 3.19).
- Otherwise, $S_1 \subseteq [B]$, and $\text{KD-FSAT}_M^k(B)$ and $\text{KD-CSAT}_M^k(B)$ are PSPACE-complete (Corollary 3.11).

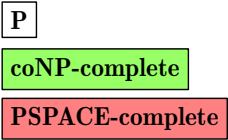
This dichotomy is a natural analog of Lewis’s result that the satisfiability problem for Boolean formulas with connectives from B is NP-complete if $S_1 \subseteq [B]$ and in P otherwise [Lew79].

From these theorems, we conclude that using the more succinct representation of modal circuits does not increase the polynomial degree of the complexity of these satisfiability problems (for two problems A and B , we write $A \equiv_m^p B$ if $A \leq_m^p B$ and $B \leq_m^p A$).

Corollary 3.3. *Let B be a finite set of Boolean functions, $\mathcal{F} \in \{\text{K}, \text{KD}\}$, $k \geq 1$, and let $M \subseteq \{\Box, \Diamond\}$. Then $\mathcal{F}\text{-CSAT}_M^k(B) \equiv_m^p \mathcal{F}\text{-FSAT}_M^k(B)$.*

The following is our classification for the logics T and S4, which gives a complete classification except for the cases where $[B]$ is one of the clones L or L_0 .

Theorem 3.4. *Let B be a finite set of Boolean functions, $\mathcal{F} \in \{\text{T}, \text{S4}\}$, $k \geq 1$, and $\emptyset \neq M \subseteq \{\Box, \Diamond\}$.*



12

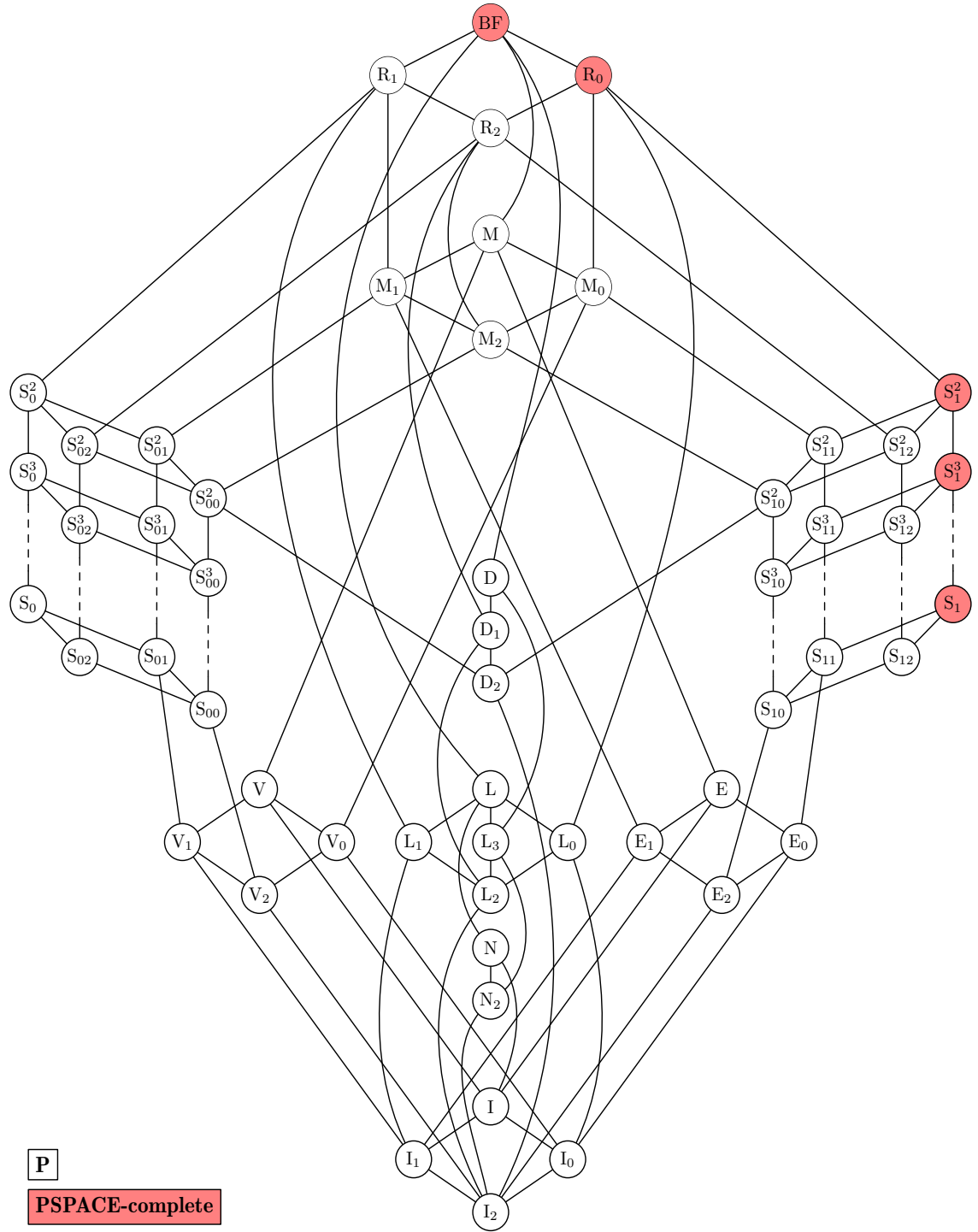


Fig. 3. The complexity of $\text{KD-FSAT}_M^k(B)$ and $\text{KD-CSAT}_M^k(B)$ for any $\emptyset \neq M \subseteq \{\square, \diamond\}$ and $\text{K-FSAT}_\diamond^k(B)$, $\text{K-FSAT}_\square^k(B)$, $\text{K-CSAT}_\diamond^k(B)$, and $\text{K-CSAT}_\square^k(B)$.

- If $B \subseteq R_1, D, N$ or M , then $\mathcal{F}\text{-CSAT}_M^k(B) \in P$ (Corollary 3.15, Theorem 3.16, Theorem 3.17)
- If $S_1 \subseteq [B]$, then $\mathcal{F}\text{-CSAT}_M^k(B)$ is PSPACE-complete.
- Otherwise, $[B] \in \{L, L_0\}$.

The logic S5 behaves differently: It is well known that the satisfiability problem for this logic can be solved in NP, as long as only one modality is present [Lad77]. As soon as at least two modalities are involved, the problem becomes PSPACE-complete [HM92]. We show that, in a similar way to the other logics with PSPACE-complete satisfiability problems that we considered, the problem is hard for this complexity class as soon as the propositional functions we allow in the formulas and circuits can express the crucial function $x \wedge \bar{y}$, which corresponds to clones that are supersets of S_1 .

Theorem 3.5. *Let B be a finite set of Boolean functions, $k \geq 1$, and $\emptyset \neq M \subseteq \{\Box, \Diamond\}$. Then the following holds:*

- If $B \subseteq R_1, D, N$ or M , then $S5\text{-CSAT}_M^k(B) \in P$ (Corollary 3.15, Theorem 3.16, Theorem 3.17)
- If $S_1 \subseteq [B]$, then $S5\text{-CSAT}_M^k(B)$ is PSPACE-complete if $k \geq 2$, and NP-complete if $k = 1$ (Corollary 3.11).
- Otherwise, $[B] \in \{L, L_0\}$.

The above classifications leave open the cases where the set B generates one of the clones L and L_0 . We will discuss these open issues in Section 3.4. Note that in the above theorem, the NP-hardness results are immediate from the previously mentioned results in [Lew79]: It directly follows from his result that for any non-empty class \mathcal{F} of frames, the problem $\mathcal{F}\text{-FSAT}_\emptyset^0(B)$ is NP-hard if $S_1 \subseteq [B]$.

The rest of this section is devoted to proving these theorems. As mentioned before, it suffices to prove upper bounds for circuits and lower bounds for formulas.

3.1 General Upper Bounds

It is well known that the \mathcal{F} -satisfiability problem for modal formulas using the operators \Box, \wedge , and \neg is solvable in PSPACE for a variety of classes \mathcal{F} of frames for both the uni-modal case [Lad77] and the general multi-modal setting [HM92]. The following theorem shows that the circuit case can be reduced to the formula case, thus putting the circuit problems in PSPACE as well.

The intuitive reason why the complexity of our satisfiability problems does not increase significantly when considering circuits instead of formulas is that for many algorithms in modal logic, the complexity depends on the number of appearing subformulas more than on the length of the formula.

Theorem 3.6. *Let B be a finite set of Boolean functions, $\mathcal{F} \in \{K, KD, T, S4, S5\}$, $k \geq 1$, and $M \subseteq \{\Box, \Diamond\}$. Then $\mathcal{F}\text{-CSAT}_M^k(B) \in \text{PSPACE}$ and $S5\text{-CSAT}_M^1(B) \in \text{NP}$.*

Proof. The main idea of the proof is to transform the given circuit in $\text{MCIRC}_M^k(B)$ into a modal formula using modal operators \Box_1, \dots, \Box_k , the modal operator E (where $E\varphi$ is an abbreviation for $\Box_1\varphi \wedge \dots \wedge \Box_k\varphi$), and the propositional symbols \wedge, \vee, \neg . Satisfiability for these formulas for the classes \mathcal{F} of frames that we consider can be solved in PSPACE and the case where $\mathcal{F} = \text{S5}$ and $k = 1$ can be solved in NP [Lad77, HM92]. Note that their proofs do not cover the E -operator, but they work without any change if $E\varphi$ is always locally evaluated as its expansion $\Box_1\varphi \wedge \dots \wedge \Box_k\varphi$ in the algorithms presented in [HM92].

The reduction works as follows: Let C be a circuit in $\text{MCIRC}_M^k(B)$ modal B -circuit with up to k modalities. Due to Lemma 2.6 and since PSPACE is closed under \leq_m^{\log} -reductions, we can without loss of generality assume that $B = \{\wedge, \neg\}$. For every gate g in C , define $f'(C, g)$ as follows:

- If g is an input gate labeled x_i , then $f'(C, g) = g \leftrightarrow x_i$.
- If g is a \neg -gate, then $f'(C, g) = g \leftrightarrow \neg h$, where h is the predecessor gate of g in C .
- If g is an \wedge -gate, then $f'(C, g) = g \leftrightarrow (h_1 \wedge h_2)$, where h_1, h_2 are the predecessor gates of g in C .
- If g is a \Box_i -gate for some $1 \leq i \leq k$, then $f'(C, g) = g \leftrightarrow \Box_i h$, where h is the predecessor gate of g in C .

In this way, the gates of the circuit are represented by variables in the corresponding formula. We will view $f'(C, g)$ as a formula over $\{\Box_1, \dots, \Box_k, \wedge, \neg\}$, by viewing “ $\varphi \leftrightarrow \psi$ ” as shorthand for “ $\neg(\varphi \wedge \neg\psi) \wedge \neg(\neg\varphi \wedge \psi)$.” Clearly, f' is computable in logarithmic space (note that the \leftrightarrow symbols do not occur nested). We now define the actual reduction as follows: For every circuit $C \in \text{MCIRC}_M^k(\wedge, \neg)$ with output gate g_{out} ,

$$f(C) = g_{\text{out}} \wedge \bigwedge_{g \text{ gate in } C} \bigwedge_{i=0}^{md(C)} E^i f'(C, g).$$

Here $E^i\varphi$ denotes $\underbrace{E \dots E}_{i \text{ times}} \varphi$. Clearly, f is computable in logarithmic space. We

will now show that C is \mathcal{F}_k -satisfiable if and only if $f(C)$ is \mathcal{F}_k -satisfiable.

First suppose that C is \mathcal{F}_k -satisfiable. Let $M = (W, R_1, \dots, R_k, \pi)$ be an \mathcal{F}_k -model, and let $w_0 \in W$ be a world such that $M, w_0 \models C$. The model M' is defined over the same set of worlds with the same successor relations, and inherits the truth assignment from M for all variables appearing in C . For the new variables, the truth assignment π' of M' is defined as follows: For every gate g in C , $\pi'(g) = \{w \in W \mid M, w \models C_g\}$. Here C_g is the subcircuit of C with output gate g . By definition of π' , for every world $w \in W$ and for every gate $g \in C$, $M', w \models g$ if and only if $M', w \models C_g$. It is easy to show (see below) that for every world $w \in W$ and for every gate $g \in C$, $M', w \models f'(C, g)$. This implies that $M', w_0 \models \bigwedge_{g \text{ gate in } C} \bigwedge_{i=0}^{md(C)} E^i f'(C, g)$. Since $M, w_0 \models C$ and $C = C_{g_{\text{out}}}$, it follows by the definition of π' that $M', w_0 \models g_{\text{out}}$. It follows that $M', w_0 \models f(C)$, and thus $f(C)$ is \mathcal{F} -satisfiable.

To be complete, we will show that, as mentioned above, for every world $w \in W$ and for every gate $g \in C$, $M', w \models f'(C, g)$. We make a case distinction.

- g is an input gate x_i . By definition of π' , $M', w \models g$ if and only if $M', w \models x_i$. It follows that $M', w \models g \leftrightarrow x_i$.
- g is a \neg -gate. Let h be the predecessor gate of g . $M', w \models g$ if and only if $M', w \models C_g$. The latter holds if and only if $M', w \not\models C_h$. This holds if and only if $M', w \not\models h$. It follows that $M', w \models g \leftrightarrow \neg h$.
- g is an \wedge -gate. Let h_1 and h_2 be the predecessor gates of g . $M', w \models g$ if and only if $M', w \models C_g$. The latter holds if and only if $M', w \models C_{h_1}$ and $M', w \models C_{h_2}$. By definition of π' , $M', w \models C_{h_1}$ if and only if $M', w \models h_1$ and $M', w \models C_{h_2}$ if and only if $M', w \models h_2$. It follows that $M', w \models g \leftrightarrow (h_1 \wedge h_2)$.
- g is a \Box_i -gate for some i . Let h be the predecessor gate of g . $M', w \models g$ if and only if $M', w \models C_g$. The latter holds if and only if $(\forall w' \in W)[wR_i w' \Rightarrow M', w' \models C_h]$. This holds if and only if $(\forall w' \in W)[wR_i w' \Rightarrow M', w' \models h]$. It follows that $M', w \models g \leftrightarrow \Box_i h$.

For the converse, suppose that $f(C)$ is \mathcal{F} -satisfiable. Let M be an \mathcal{F} -model, and let $w_0 \in W$ be a world such that $M, w_0 \models f(C)$. We will prove by induction on the structure of circuit C_g that for every gate $g \in C$ and for every world w that is reachable from w_0 in at most $md(C) - md(C_g)$ steps, $M, w \models C_g$ if and only if $M, w \models g$. This clearly implies that $M, w_0 \models C$, and thus C is \mathcal{F} -satisfiable.

- g is an input gate x_i . Then C_g is equivalent to x_i . Since $M, w \models g \leftrightarrow x_i$, it follows that $M, w \models C_g$ if and only if $M, w \models g$.
- g is a \neg -gate. Let h be the predecessor gate of g . Then $M, w \models C_g$ if and only if $M, w \not\models C_h$. By induction, the latter holds if and only if $M, w \not\models h$. Clearly, $M, w \not\models h$ if and only if $M, w \models \neg h$. Since $M, w \models g \leftrightarrow \neg h$, it follows that $M, w \models C_g$ if and only if $M, w \models g$, as required.
- g is an \wedge -gate. Let h_1 and h_2 be the predecessor gates of g . Then $M, w \models C_g$ if and only if $M, w \models C_{h_1}$ and $M, w \models C_{h_2}$. By induction, the latter holds if and only if $M, w \models h_1$ and $M, w \models h_2$, and this holds if and only if $M, w \models h_1 \wedge h_2$. Since $M, w \models g \leftrightarrow (h_1 \wedge h_2)$, it follows that $M, w \models C_g$ if and only if $M, w \models g$, as required.
- g is a \Box_i -gate for some i . Let h be the predecessor gate of g . Then $M, w \models C_g$ if and only if for all $w' \in W$ such that $wR_i w'$, it holds that $M, w' \models C_h$. Note that $md(C_h) = md(C_g) - 1$. Since w is reachable from w_0 in at most $md(C) - md(C_g)$ steps, it follows that for every w' such that $wR_i w'$, w' is reachable from w_0 in at most $md(C) - md(C_g) + 1 = md(C) - md(C_h)$ steps. And so, by induction, it follows that (for all $w' \in W$ such that $wR_i w'$, it holds that $M, w' \models C_h$) if and only if (for all $w' \in W$ such that $wR_i w'$, it holds that $M, w' \models h$), and this holds if and only if $M, w \models \Box_i h$. Since $M, w \models g \leftrightarrow \Box_i h$, it follows that $M, w \models C_g$ if and only if $M, w \models g$, as required.

Finally note that the KD case easily follows from the result for K, since a circuit C is KD-satisfiable if and only if $C \wedge \bigwedge_{i=0}^{md(\varphi)} E^i \bigwedge_{j=1}^k \Diamond_j 1$ is K-satisfiable. \square

Note that in the uni-modal case, we do not have to introduce the E -operator as in the proof above. Therefore the construction of the proof directly implies that for any class \mathcal{F} of frames, uni-modal satisfiability for circuits (using any set of propositional gates) is not more difficult than the satisfiability problem for $\{\wedge, \neg\}$ -formulas for the same class of frames.

Corollary 3.7. *Let B be a finite set of Boolean functions and \mathcal{F} a class of frames. Then $\mathcal{F}\text{-CSAT}_{\square, \diamond}^1(B) \leq_m^p \mathcal{F}\text{-FSAT}_{\square}^1(\wedge, \neg)$.*

3.2 PSPACE-completeness

We now show how to express, in a satisfiability-preserving way, uni-modal formulas and circuits using a restricted set of Boolean connectives and one modal operator. This implies that our satisfiability problems for these restricted sets of formulas are as hard as the general case.

As mentioned in the discussion following Lemma 2.6, with many formula transformations, the size of the resulting formula can be exponential. A crucial tool in dealing with this situation is the following lemma showing that for certain sets B , there are always short formulas representing the functions AND, OR, and NOT. Part (1) is Lemma 1.4.5 from [Sch07], the result for the case $[B] = \text{BF}$ is proven in [Lew79]. Part (2) follows directly from the proofs in [Lew79].

Lemma 3.8. *Let B be a finite set of Boolean functions.*

1. *If $V \subseteq [B]$ ($E \subseteq [B]$, resp.), then there exists a B -formula $f(x, y)$ such that f represents $x \vee y$ ($x \wedge y$, resp.) and each of the variables x and y occurs exactly once in $f(x, y)$.*
2. *If $N \subseteq [B]$, then there exists a B -formula $f(x)$ such that f represents \bar{x} and the variable x occurs in f only once.*

The proof of the following theorem uses a generalization of ideas from the proof for the main result in [Lew79]. This can be applied to an arbitrary class of frames, and in particular, it yields PSPACE completeness results for K and KD .

Theorem 3.9. *Let B be a finite set of Boolean functions such that $S_1 \subseteq [B]$, \mathcal{F} a class of frames, and $\emptyset \neq M \subseteq \{\square, \diamond\}$. Then the following holds:*

- $\mathcal{F}\text{-FSAT}_{\square, \diamond}^1(\wedge, \neg) \leq_m^{\log} \mathcal{F}\text{-FSAT}_M^1(B)$,
- $\text{S5-FSAT}_{\square, \diamond}^2(\wedge, \neg) \leq_m^{\log} \text{S5-FSAT}_M^2(B)$.

Proof. First consider the uni-modal case. Let $\varphi \in \text{MFORM}_{\square, \diamond}^1(\wedge, \neg)$. Without loss of generality, assume that φ contains only modal operators from M (use the identity $\square \equiv \neg \diamond \neg$ otherwise). Let $B' := B \cup \{1\}$. Then Figure 1 shows that $[B'] = \text{BF}$ (since I_1 is the smallest clone containing 1, and BF is the smallest clone containing I_1 and S_1). It follows from Lemma 3.8 that there is a B' -formula $f_{\neg}(x)$ that represents \bar{x} , and x occurs in $f_{\neg}(x)$ only once, and there exist B' -formulas $f_{\wedge}(x, y)$ and $f_{\vee}(x, y)$ such that f_{\wedge} represents \wedge , $f_{\vee}(x, y)$ represents \vee ,

and x and y occur exactly once in $f_\wedge(x, y)$ and exactly once in $f_\vee(x, y)$. In φ , replace every occurrence of \wedge with f_\wedge , every occurrence of \vee with f_\vee , and every occurrence of \neg with f_\neg . Call the resulting formula φ' . Clearly, φ' is a formula in $\text{MFORM}_M^1(B')$, and φ' is equivalent to φ . By choice of f_\vee , f_\wedge , and f_\neg , φ' is computable in polynomial time.

Now replace every occurrence of the constant 1 with a new variable t and force t to be 1 in every relevant world by adding $\bigwedge_{i=0}^{md(\varphi)} \Box_1^i t$. This is a conjunction of linearly many terms (since $md(\varphi) \leq |\varphi|$). We insert parentheses in such a way that we get a tree of \wedge 's of logarithmic depth. Now express the \wedge 's in this tree with the equivalent B -formula (which exists, since $[B] \supseteq S_1 \supseteq E_2 = [\wedge]$) with the result only increasing polynomially in size. It is obvious that this formula is satisfiable if and only if the original formula is.

Now for the bimodal case and the logic S5, we use the same construction as above, except that to force the variable t to true in all relevant worlds, we use the formula $(\Box_1 \Box_2)^{md(\varphi)} t$. Due to the reflexivity of both successor relations in S5₂-models, this forces t to be true in all relevant worlds. \square

The following theorem implies that for the logic K, PSPACE-completeness already holds for a lower class in Post's Lattice. The proof is nearly identical to the one for the above Theorem 3.9: Note that $[S_{11} \cup \{1\}] = M$, and apply Lemma 3.8 for the class M. Then follow the construction above. (We can represent \wedge by a B -formula since $S_{11} \supseteq E_2 = [\wedge]$, and we can represent 0 by a B -formula since $0 \in S_{11}$.)

Theorem 3.10. *Let B be a finite set of Boolean functions such that $S_{11} \subseteq [B]$, \mathcal{F} a class of frames, $k \geq 1$, and $M \subseteq \{\Box, \Diamond\}$. Then $\mathcal{F}\text{-FSAT}_M^1(\wedge, \vee, 0) \leq_m^{\log} \mathcal{F}\text{-FSAT}_M^1(B)$.*

The above theorems give the following corollary.

Corollary 3.11. *Let B be a finite set of Boolean functions, and let $\emptyset \neq M \subseteq \{\Box, \Diamond\}$.*

1. *If $[B] \supseteq S_1$, and \mathcal{F} is a class of frames such that $S_4 \subseteq \mathcal{F} \subseteq K$, and $k \geq 1$, then $\mathcal{F}\text{-FSAT}_M^k(B)$ and $\mathcal{F}\text{-CSAT}_M^k(B)$ are PSPACE-hard.*
2. *If $[B] \supseteq S_{11}$ and $k \geq 1$, then $K\text{-FSAT}_{\Box, \Diamond}^k(B)$ and $K\text{-CSAT}_{\Box, \Diamond}^k(B)$ are PSPACE-complete.*
3. *If $[B] \supseteq S_1$ and $k \geq 2$, then $S5\text{-FSAT}_M^k(B)$ and $S5\text{-CSAT}_M^k(B)$ are PSPACE-complete.*

Proof. The upper bounds follow from Theorem 3.6.

1. In [Lad77], it is shown that for every class of frames \mathcal{F} such that $S_4 \subseteq \mathcal{F} \subseteq K$, the problem $\mathcal{F}\text{-FSAT}_M^1(\wedge, \neg)$ is PSPACE-hard. Therefore this follows from [Lad77] and Theorem 3.9.
2. In [Hem01, Theorem 6.5], it is shown that $K\text{-FSAT}_{\Box, \Diamond}^1(\wedge, \vee, 0)$ is PSPACE-hard. Thus the result follows from Theorem 3.10.
3. In [HM92], it is shown that $S5\text{-FSAT}_{\Box, \Diamond}^2(\wedge, \neg)$ is PSPACE-hard. Therefore, the result follows from Theorem 3.9. \square

3.3 coNP-completeness

In [Hem01], the analogous result of the following lemma was shown for uni-modal formulas. We prove that this coNP upper bound also holds for circuits.

Lemma 3.12. *Let $k \geq 1$. Then $\text{K-CSAT}_{\square, \diamond}^k(\wedge, 0, 1) \in \text{coNP}$.*

Proof. The proof for the analogous statement for uni-modal formulas is based on the following fact: Let φ be a formula of the form $\varphi = \bigwedge_{i \in I} \square \varphi_i^\square \wedge \bigwedge_{j \in J} \diamond \varphi_j^\diamond \wedge \psi$, where I and J are finite sets of indices, φ_i^\square and φ_j^\diamond are modal formulas for all $i \in I$, $j \in J$, and ψ is a propositional formula. Then φ is satisfiable if and only if ψ is satisfiable and for every $j \in J$, $\bigwedge_{i \in I} \varphi_i^\square \wedge \varphi_j^\diamond$ is satisfiable [Lad77].

This generalizes to multi-modal formulas from $\text{MFORM}_{\square, \diamond}^k(\wedge, 0, 1)$ in the following way: let

$$\varphi = \bigwedge_{i \in I_1} \square_1 \varphi_i^{\square_1} \wedge \cdots \wedge \bigwedge_{i \in I_k} \square_k \varphi_i^{\square_k} \wedge \bigwedge_{j \in J_1} \diamond_1 \varphi_j^{\diamond_1} \wedge \cdots \wedge \bigwedge_{j \in J_k} \diamond_k \varphi_j^{\diamond_k} \wedge \psi,$$

for finite sets of indices $I_1, \dots, I_k, J_1, \dots, J_k$, formulas $\varphi_i^{\square_l}, \varphi_j^{\diamond_l} \in \text{MFORM}_{\square, \diamond}^k(\wedge, 0, 1)$, and a propositional $\{\wedge, 0, 1\}$ -formula ψ . Then φ is satisfiable if and only if for every $1 \leq l \leq k$ and every $j \in J_l$ it holds that ψ and $\bigwedge_{i \in I_l} \varphi_i^{\square_l} \wedge \varphi_j^{\diamond_l}$ are satisfiable. Since every formula from $\text{MFORM}_{\square, \diamond}^k(\wedge, 0, 1)$ can be written in the above form and since satisfiability for the propositional part ψ can be tested in polynomial time according to [Lew79], this leads to a recursive NP-algorithm for the question if φ is unsatisfiable.

We give an analogous proof for multi-modal circuits. Let C be a circuit from $\text{MCIRC}_{\square, \diamond}^k(\wedge, 0, 1)$ with output-gate out . If out is a \square_i -gate for some $1 \leq i \leq k$, then φ is satisfied in every world without a successor, if out is a \diamond_i -gate for some $1 \leq i \leq k$, then C is satisfiable if and only if the circuit obtained from C by using the predecessor of out as output-gate is satisfiable, and finally if out is an input-gate or a constant gate, then satisfiability can be tested trivially. Therefore we assume without loss of generality out to be an \wedge -gate. For a set of gates G we define $\text{pred}(G)$ to be the set of all direct predecessor gates of gates in G and $\wedge\text{-pred}(G)$ to be the set of all non \wedge -gates g which are connected to G by a path from g to a gate $g' \in G$ where all gates on the path excluding g (but including g' if $g \neq g'$) are \wedge -gates.

For $1 \leq i \leq k$ let G_{\square_i} be the set of all \square_i -gates in C , G_{\diamond_i} the set of all \diamond_i -gates in C and G the set of all propositional gates in C . Then, due to the equivalence above, C is satisfiable if and only if

$$\bigwedge_{g \in \wedge\text{-pred}(\{out\}) \cap G} \varphi_g \quad \text{and} \quad \bigwedge_{g \in \text{pred}(\wedge\text{-pred}(\{out\}) \cap G_{\square_i})} \varphi_g \wedge \bigwedge_{g \in \text{pred}(\{g_{\diamond_i}\})} \varphi_g$$

are satisfiable for every $1 \leq i \leq k$ and every $g_{\diamond_i} \in \wedge\text{-pred}(\{out\}) \cap G_{\diamond_i}$, where for a gate g , the formula φ_g is defined as in the definition for modal circuits, i.e., φ_g is the formula represented by the sub-circuit with output-gate g . Note

that due to the definition of \wedge -pred, the first of these formulas is a propositional formula.

More generally, a formula of the form $\varphi = \bigwedge_{g \in H} \varphi_g$ for a set H of gates from C is satisfiable if and only if

$$\psi := \bigwedge_{g \in \wedge\text{-pred}(H) \cap G} \varphi_g \quad \text{and} \quad \varphi^{g_{\diamond_i}} := \bigwedge_{g \in \text{pred}(\wedge\text{-pred}(H) \cap G_{\square_i})} \varphi_g \wedge \bigwedge_{g \in \text{pred}(\{g_{\diamond_i}\})} \varphi_g$$

are satisfiable for every $1 \leq i \leq k$ and every $g_{\diamond_i} \in \wedge\text{-pred}(H) \cap G_{\diamond_i}$.

Note that ψ is a conjunction of constants and variables, therefore satisfiability of ψ can be tested in polynomial time. It is obvious that constructing the sets $\text{pred}(H)$ and $\wedge\text{-pred}(H)$ needs only polynomial time as well.

For testing if a formula φ represented by H is unsatisfiable it suffices to check if ψ is unsatisfiable, and, if this is not the case, to guess a $g_{\diamond_i} \in \wedge\text{-pred}(H) \cap G_{\diamond_i}$ for some $1 \leq i \leq k$ and to recursively test unsatisfiability of $\varphi^{g_{\diamond_i}}$, which is represented by the set $\text{pred}(\wedge\text{-pred}(H) \cap G_{\square_i}) \cup \text{pred}(\{g_{\diamond_i}\})$. Since in every recursion the length of the longest path between an input-gate and a gate in H decreases, the algorithm stops after at most $|C|$ recursions.

Hence, starting with $H = \{\text{out}\}$ we get an NP-algorithm for testing unsatisfiability of C . \square

In [Hem05], it is shown that $\text{K-FSAT}_{\square, \diamond}^1(\wedge, 0)$ is coNP-hard. Applying Lemma 3.8, we obtain the following result.

Lemma 3.13. *Let B be a finite set of Boolean functions such that $E \supseteq [B] \supseteq E_0$, and $k \geq 1$. Then $\text{K-FSAT}_{\square, \diamond}^k(B)$ is coNP-hard.*

Proof. It obviously suffices to consider the case $k = 1$. We use a similar construction as in the proof for Theorem 3.9. Let $B' := B \cup \{1\}$. From the structure of Post's Lattice, it follows that $[B'] = E$. Hence, by Lemma 3.8, we have a short B' -formula for AND, and can convert $\text{MFORM}_{\square, \diamond}^1(\wedge, 0)$ -formulas into equivalent formulas from $\text{MFORM}_{\square, \diamond}^1(B')$. We remove the occurrences of 1 as in Theorem 3.9: Introduce a variable t and force it to be 1 with the logarithmic tree construction. The coNP-hardness then follows from the above-mentioned result from [Hem05]. \square

3.4 Polynomial Time

We now give our polynomial-time algorithms. We will see that in many of those cases where the restriction of the propositional operators to a certain set B leads to a polynomial-time decision procedure in the propositional case, the same is true for the corresponding modal problems. One notable exception is the case of monotone formulas: For propositional monotone formulas, satisfiability can easily be tested, since such a formula is satisfiable if and only if it is satisfied by the constant 1-assignment. For modal satisfiability, we have seen in Corollary 3.11 that the corresponding problem is as hard as the standard satisfiability problem

for modal logic. The other exception concerns formulas using only conjunction and constants: As a special case of monotone formulas, satisfiability testing is easy for propositional logic. However, Section 3.3 showed that the problem is coNP-complete for modal logic.

Lemma 3.14. *Let B be a finite set of Boolean functions, $k \geq 1$, and $\varphi \in \text{MFORM}_{\Box, \Diamond}^k(B)$. If the formula φ^{id} , which is obtained by changing every modal operator in φ to the identity, is satisfiable, then φ is satisfiable in the reflexive singleton.*

Proof. Let I be a propositional assignment satisfying φ^{id} . Let M be the model consisting of the reflexive singleton, where each variable is true if and only if it is true in I . Since in this model, every modal operator can only refer to the same single world in the model, the operators are equivalent to the identity function, implying the result. \square

It is obvious that every propositional B -formula for $B \subseteq R_1$ or $B \subseteq D$ is satisfiable ([Lew79]): In the first case, the all-1-assignment always is a model. In the second case, exactly one of the two constant assignments is. Hence, Lemma 3.14 immediately gives the following complexity result:

Corollary 3.15. *Let B be a finite set of Boolean functions such that $B \subseteq R_1$ or $B \subseteq D$, \mathcal{F} a class of frames containing the reflexive singleton, and $k \geq 1$. Then every formula from $\text{MFORM}_{\Box, \Diamond}^k(B)$ is \mathcal{F} -satisfiable. In particular, $\mathcal{F}\text{-CSAT}_{\Box, \Diamond}^k(B) \in \text{P}$ for $\mathcal{F} \in \{K, KD, K4, T, S4, S5\}$.*

While K-satisfiability for variable-free formulas using constants, the Boolean connectives \wedge and \vee , and both modal operators is complete for PSPACE [Hem01], this problem (even with variables) is solvable in polynomial time if we look only at frames in which each world has a successor.

Theorem 3.16. *Let B be a finite set of Boolean functions such that $B \subseteq M$, \mathcal{F} a class of frames such that $\mathcal{F} \subseteq KD$, and $k \geq 1$. Then $\mathcal{F}\text{-CSAT}_{\Box, \Diamond}^k(B) \in \text{P}$. In particular, $KD\text{-CSAT}_{\Box, \Diamond}^k(B)$, $T\text{-CSAT}_{\Box, \Diamond}^k(B)$, $S4\text{-CSAT}_{\Box, \Diamond}^k(B)$, $S5\text{-CSAT}_{\Box, \Diamond}^k(B) \in \text{P}$.*

Proof. The claim is obvious if \mathcal{F} is empty, hence assume that this is not the case. Let M be an \mathcal{F} -model, let w be a world from M , and let M_1 be the multi-modal reflexive singleton with k successor relations in which every variable is set to 1. It is easy to show by induction on the construction of any $C \in \text{MCIRC}_M^k(B)$ that if $M, w \models C$, then $M_1, w \models C$ holds as well. On the other hand, if $M_1, w \models C$, then $M', w \models C$, where M' is obtained from the model M by setting every variable to true in every world. Hence, C is \mathcal{F} -satisfiable if and only if C is satisfied in M_1 . The latter condition can obviously be verified in polynomial time. \square

In the case where all of our propositional operators are unary, we can use simple transformations to decide satisfiability, as the following theorem shows.

Theorem 3.17. *Let B be a finite set of Boolean functions such that $B \subseteq N$, \mathcal{F} a class of frames such that $\mathcal{F} \in \{K, KD, S4, S5, K4, T\}$, and $k \geq 1$. Then $\mathcal{F}\text{-CSAT}_{\Box, \Diamond}^k(B) \in P$.*

Proof. Since the clone N is generated by negation and the constants, we can, due to Lemma 2.6, assume that B only contains these functions.

Now, let B be a circuit from $\text{MCIRC}_M^k(B)$. Since every function in B is unary or constant, C is a linear graph, and we can therefore regard C as a formula. Using the equivalence $\Diamond_i \equiv \neg \Box_i \neg$, we can move negations inward, until we have a formula of the form $O_1 \dots O_n z$, where the O_i are modal operators, and z is either a literal or a constant. It is obvious that this formula is satisfiable if and only if z is not the constant 0, or if $\mathcal{F} = K$, and there is at least one \Box -operator present. The transformation obviously can be performed in polynomial time. \square

For monotone functions and most classes of frames that we are interested in, we already showed that the satisfiability problem can be solved in polynomial time. For the most general class of frames K , this problem is PSPACE-complete (Corollary 3.11), but a further restriction of the propositional base gives polynomial-time results here as well.

Theorem 3.18. *Let B be a finite set of Boolean functions such that $B \subseteq V$, \mathcal{F} a class of frames such that $\mathcal{F} \in \{K, KD, S4, S5, K4, T\}$, and $k \geq 1$. Then $\mathcal{F}\text{-CSAT}_{\Box, \Diamond}^k(B) \in P$.*

Proof. Since the clone V is generated by binary OR and the constants, we can, due to Lemma 2.6, assume that B only contains these functions. We first consider the case $\mathcal{F} \in \{K, K4\}$.

Let B be a circuit from $\text{MCIRC}_{\Box, \Diamond}^k(B)$. If the output gate g of C is an \vee -gate, with predecessors h_1 and h_2 in C , then C is \mathcal{F} -satisfiable if and only if at least one of C_{h_1} and C_{h_2} is. If g is a \Diamond_i -gate with predecessor h , then C is \mathcal{F} -satisfiable if and only if C_h is. Finally, if g is a \Box_i -gate, then C is K -satisfiable.

This gives a recursive polynomial-time procedure to decide the satisfiability problem. For the classes other than K and $K4$, we can use the same procedure, with one exception: here, if g is a \Box_i -gate, then C is satisfiable if and only if C_h is satisfiable, where h is the predecessor of g in C . \square

We now show that for the logics K and KD , the modal satisfiability problems for formulas having only \oplus and constants in the propositional base are easy. For the propositional case, this holds because unsatisfiable formulas using only these connectives are of a very easy form: Every variable and the constant 1 appear an even number of times (see, e.g., [Lew79]). In the modal case, unsatisfiable formulas over these connectives are of a similarly regular form, as we will soon see. The result also holds for modal circuits.

Theorem 3.19. *Let B be a finite set of Boolean functions such that $B \subseteq L$, $\mathcal{F} \in \{K, KD\}$ a class of frames, and $k \geq 1$. Then $\mathcal{F}\text{-CSAT}_{\Box, \Diamond}^k(B) \in P$.*

To prove this theorem, we present a polynomial-time algorithm deciding the problem. Because of Lemma 2.6, we can restrict ourselves to circuits from $\text{MCIRC}_{\square, \diamond}^k(\oplus, 0, 1)$. First note that using \diamond_i, \oplus , and the constant 1, we can express \square_i , and therefore it is sufficient to consider circuits in which only \diamond_i -operators occur, i.e., we only need to deal with circuits from $\text{MCIRC}_{\diamond}^k(\oplus, 0, 1)$.

The algorithm \oplus -SAT presented below decides this problem in polynomial time by converting circuits into a normal form. For a circuit C , let $\oplus\text{-SAT}(C)$ denote the output of the algorithm \oplus -SAT when given C as input. A decision algorithm derived from \oplus -SAT accepts a circuit C if and only if $\oplus\text{-SAT}(C)$ is not the constant 0-circuit.

The intuitive approach of the algorithm is to delete redundant data, i.e., extra 0s and sub-circuits corresponding to formulas of the form $\varphi \oplus \varphi$, which obviously are equivalent to 0, and to arrange the gates of the circuit in a standard order, to get a unique representation for the input circuit. In the propositional formula case, the approach is quite simple: For a formula in which only the operator \oplus , variables and constants appear, we repeatedly delete every variable or constant that appears twice, and remove 0s. If this produces the empty formula or the formula containing only the constant 0, then the formula is unsatisfiable, otherwise it is satisfiable. Surprisingly, the generalization to modal logic and circuits instead of formulas performs only operations of a similarly simple type—however, proving the correctness requires more work than in the propositional case.

In the statement of the algorithm, the term \diamond -gate refers to any \diamond_i -gate for some $i \in \{1, \dots, k\}$.

$\oplus\text{-SAT}(\text{Input: } C \in \text{MCIRC}_{\diamond}^k(\{\oplus, 0, 1\}))$
while there are unmarked \diamond -gates or the output gate is not marked **do**
 Let g be an unmarked \diamond -gate such that all \diamond -gates with a path to g are marked if such a gate exists, let g be the output gate otherwise.
 Let G be the set of propositional gates before g which are connected to g with a path consisting only of propositional gates (G includes g if g is propositional).
 Let D_1, \dots, D_m be the subcircuits whose output gates are the \diamond -gates directly before G .
 Consider G as a propositional circuit with output gate g and input gates d_1, \dots, d_m replacing the subcircuits D_1, \dots, D_m .
 Rewrite G as formula $\varphi := d_{i_1} \oplus d_{i_2} \oplus \dots \oplus d_{i_j} \oplus \varphi'$, where each d_i occurs at most once and where φ' does not contain d_1, \dots, d_m .
 while changes in φ still occur **do**
 Order φ lexicographically.
 If D_i and D_j are identical, replace $d_i \oplus d_j$ with 0.
 If $\mathcal{F} = \text{KD}$, then replace $\diamond_i 1$ with 1 for any i .
 For any i , replace $\diamond_i 0$ with 0.

Remove 0s unless the formula becomes empty.
 For propositional variable p , replace $p \oplus p$ with 0.
 Replace $1 \oplus 1$ with 0.

end while

Reintegrate φ into the circuit, using connections from the D_i subcircuits instead of the d_i variables.

mark g

end while

Delete gates not connected to the output gate.

We now show that the algorithm works correctly—note that the following theorem implies the correctness of the decision procedure outlined above, since $\oplus\text{-SAT}$ returns 0 when given a circuit consisting just of a 0-gate as input.

Theorem 3.20. *Let C_1 and C_2 be circuits from $\text{MCIRC}_{\Diamond}^k(\{\oplus, 0, 1\})$, and let $\mathcal{F} \in \{\text{K}, \text{KD}\}$. Then $\oplus\text{-SAT}(C_1) = \oplus\text{-SAT}(C_2)$ if and only if $C_1 \equiv_{\mathcal{F}} C_2$.*

First we show that the algorithm can be implemented to work in polynomial time, and observe a useful property.

Lemma 3.21. *The algorithm $\oplus\text{-SAT}$ runs in polynomial time and satisfies $\oplus\text{-SAT}(\oplus\text{-SAT}(C)) = \oplus\text{-SAT}(C)$ for every circuit $C \in \text{MCIRC}_{\Diamond}^k(0, 1, \oplus)$ for all $k \geq 1$.*

Proof. We show that the algorithm works in polynomial time. The outer WHILE loop is run at most once for every gate in the circuit. The inner WHILE loop shortens the formula by at least one character in each iteration except one (where only sorting is performed). Each step in the algorithm can clearly be performed in polynomial time, the only non-obvious case is the “Rewrite G as formula” step. This can be performed in polynomial time because propositional circuits representing linear functions can easily be converted into formulas: Determine, by simulation, which of the variables is relevant for the function calculated by the circuit. The resulting formula consists of an XOR of all these variables and output value of the circuit when given zeros as input. Note that not all of the variables d_1, \dots, d_m necessarily appear in the formula. The formula constructed in this way is at most as large as the original circuit.

Note that if G already is a formula connected only to the output-gate g and the d_i -gates, and φ is lexicographically ordered, then the algorithm does not perform any changes at this step. This implies that $\oplus\text{-SAT}(\oplus\text{-SAT}(C)) = \oplus\text{-SAT}(C)$. \square

We now prove a lemma needed in the correctness proof for the algorithm. The lemma states that for two XOR-formulas to be equivalent, two of the arguments to the XOR operators already have to be equivalent, and this enables us to give an inductive proof for Theorem 3.20.

Lemma 3.22. Let $\mathcal{F} \in \{K, KD\}$, $k \geq 1$, $n \geq 2$, $D_1, \dots, D_n \in \text{MCIRC}_{\Diamond}^k(\{\oplus, 0, 1\})$, let φ_1, φ_2 be propositional XOR-formulas, and let $\Diamond_{i_1} D_1 \oplus \dots \oplus \Diamond_{i_n} D_n \oplus \varphi_1 \oplus \varphi_2$ be not \mathcal{F} -satisfiable, where $i_1, \dots, i_n \in \{1, \dots, k\}$. Then

1. If $\mathcal{F} = K$ and all D_i are \mathcal{F} -satisfiable, then there exist $1 \leq i \neq j \leq n$ such that $D_i \equiv_{\mathcal{F}} D_j$.
2. If $\mathcal{F} = KD$ and all D_i are \mathcal{F} -satisfiable and not \mathcal{F} -tautologies, then there exist $1 \leq i \neq j \leq n$ such that $D_i \equiv_{\mathcal{F}} D_j$.

Proof. We first show that φ_1 is equivalent to φ_2 or to $\neg\varphi_2$. Consider an \mathcal{F} -model M with a non-reflexive root world w . Changing truth assignments in w only affects the propositional formulas φ_1 and φ_2 . Since $\varphi_1 \oplus \varphi_2$ is \mathcal{F} -equivalent to $\Diamond_{i_1} D_1 \oplus \dots \oplus \Diamond_{i_n} D_n$, $\varphi_1 \oplus \varphi_2$ must be constant. This only leaves these two choices for φ_1, φ_2 .

Further, if $\mathcal{F} = K$, then $\varphi_1 \equiv \varphi_2$: Consider the frame M with a world w which does not have a successor. Since $\Diamond_{i_1} D_1 \oplus \dots \oplus \Diamond_{i_n} D_n \oplus \varphi_1 \oplus \varphi_2$ is not K -satisfiable, this implies that $\varphi_1 \oplus \varphi_2$ is not K -satisfiable, thus φ_1 and φ_2 are K -equivalent. Since these formulas are propositional, they are equivalent.

If $\varphi_1 \equiv \varphi_2$, then $\Diamond_{i_1} D_1 \oplus \dots \oplus \Diamond_{i_n} D_n$ is not \mathcal{F} -satisfiable. If $\mathcal{F} = KD$ and $\varphi_1 \equiv \neg\varphi_2$, then $\Diamond_{i_1} D_1 \oplus \dots \oplus \Diamond_{i_n} D_n$ is an \mathcal{F} -tautology. Assume that the D_l are pairwise \mathcal{F} -inequivalent. Since all of the D_l are \mathcal{F} -satisfiable, this implies that n is even for $\varphi_1 \equiv \varphi_2$, and odd for $\varphi_1 \equiv \neg\varphi_2$: If this would not hold, we could construct a world which for every D_i has an i -successor in which it holds, and this would satisfy $\Diamond_{i_1} D_1 \oplus \dots \oplus \Diamond_{i_n} D_n \oplus \varphi_1 \oplus \varphi_2$.

Let D_m be a minimal element of $\{D_1, \dots, D_n\}$ with respect to \mathcal{F} -implication. This exists because \mathcal{F} -implication defines a partial order on the D_i (the \mathcal{F} -inequivalence of the D_i ensures the anti-symmetry). For each $l \neq k$, let M_l be a model with a world w_l such that $M_l, w_l \models D_l \wedge \neg D_m$. Let M be a model containing a world w which has all of the w_l as successors. Then it holds that $M, w \models \neg\Diamond_{i_m} D_m \wedge \bigwedge_{l \neq k} \Diamond_{i_l} D_l$, and thus M, w satisfies an odd number of the $\Diamond_{i_m} D_m$ clauses if n is even, and an even number if n is odd. This model leads to a different truth value of the formula than the model where all of the $\Diamond_{i_m} D_m$'s are satisfied, which is a contradiction, because the formula is \mathcal{F} -constant. \square

We now prove Theorem 3.20:

Proof. The *propositional level* of a modal circuit C with a propositional output gate is the set of propositional gates in C that are connected to the output gate with a path having no gates representing modal operators.

Obviously, $C \equiv_{\mathcal{F}} \oplus\text{-SAT}(C)$, and therefore $\oplus\text{-SAT}(C_1) = \oplus\text{-SAT}(C_2)$ implies $C_1 \equiv_{\mathcal{F}} C_2$. We now show the other direction.

Observe that the following holds when $\oplus\text{-SAT}$ is given a circuit as input which on its propositional level is a formula (i.e., every gate in the propositional level has fan-out of at most 1), which has circuits D_i as inputs:

$$\oplus\text{-SAT}(\Diamond_{i_1} D_1 \oplus \dots \oplus \Diamond_{i_l} D_l) = \oplus\text{-SAT}(\Diamond_{i_1} \oplus\text{-SAT}(D_1) \oplus \dots \oplus \Diamond_{i_l} \oplus\text{-SAT}(D_l)) \quad (1)$$

Assume that the theorem does not hold, and let C_1, C_2 be \mathcal{F} -equivalent circuits, l_i the number of diamonds in C_i , such that $\oplus\text{-SAT}(C_1) \neq \oplus\text{-SAT}(C_2)$ and such that the pair (C_1, C_2) is minimal with respect to $l_1 + l_2$, and let $l_1 \geq l_2$. Because of Lemma 3.21, and since $\oplus\text{-SAT}$ does not add diamonds, we can assume $\oplus\text{-SAT}(C_1) = C_1$ and $\oplus\text{-SAT}(C_2) = C_2$.

If the output gate of C_1 is propositional (without loss of generality, this is an \oplus -gate), then the algorithm converts the propositional level of the circuit to a formula over the variables corresponding to the \diamond -gates which are connected to the output gates with a non-modal path. Thus, since $\oplus\text{-SAT}(C_1) = C_1$, we can consider the circuit as a formula $C_1 = \diamond_{i_1} D_1 \oplus \dots \oplus \diamond_{i_l} D_l \oplus \varphi_1$, where the D_l are the subcircuits starting before the highest diamonds. If the output gate of C_1 is modal, then C_1 is of the same form, with $k = l$ and φ_1 absent. In the same way, assume $C_2 = \diamond_{i_{l+1}} D_{l+1} \oplus \dots \oplus \diamond_{i_n} D_n \oplus \varphi_2$.

The circuits D_1, \dots, D_l are pairwise \mathcal{F} -inequivalent: Assume $D_1 \equiv_{\mathcal{F}} D_2$. Then, by minimality of C_1, C_2 , it holds that $\oplus\text{-SAT}(D_1) = \oplus\text{-SAT}(D_2)$. Therefore, because the D_j are lexicographically ordered (since $\oplus\text{-SAT}(C_1) = C_1$), equation (1) implies that $\diamond_{i_1} \oplus\text{-SAT}(D_1) \oplus \diamond_{i_2} \oplus\text{-SAT}(D_2)$ will be replaced with 0, which is a contradiction to $\oplus\text{-SAT}(C_1) = C_1$. The same holds for D_{l+1}, \dots, D_n . By an analogous argument, all of the D_j are \mathcal{F} -satisfiable: $\oplus\text{-SAT}$ converts unsatisfiable D_j to 0 and deletes them, since the D_j have less diamonds than $l_1 + l_2$. Additionally, if $\mathcal{F} = \text{KD}$, we can assume that none of the D_j is a KD-tautology, because $\diamond_{i_j} 1$ is replaced by 1.

Assume there exist i, j such that $1 \leq i \leq l < j \leq n$, and $D_i \equiv_{\mathcal{F}} D_j$. By minimality of $l_1 + l_2$, it holds that $\oplus\text{-SAT}(D_1) = \oplus\text{-SAT}(D_{l+1})$. Define E as $\oplus\text{-SAT}(D_1)$. Now, since we have

$$\begin{aligned} C_1 = \oplus\text{-SAT}(C_1) &= \diamond_{i_1} \oplus\text{-SAT}(D_1) \oplus \dots \oplus \diamond_{i_{l-1}} \oplus\text{-SAT}(D_{i_{l-1}}) \oplus \diamond_{i_l} E \\ &\quad \oplus \diamond_{i_{l+1}} \oplus\text{-SAT}(D_{l+1}) \oplus \dots \oplus \diamond_{i_l} \oplus\text{-SAT}(D_l) \oplus (\oplus\text{-SAT}(\varphi_1)) \\ C_2 = \oplus\text{-SAT}(C_2) &= \diamond_{i_{l+1}} \oplus\text{-SAT}(D_{l+1}) \oplus \dots \oplus \diamond_{i_{j-1}} \oplus\text{-SAT}(D_{j-1}) \oplus \diamond E \\ &\quad \oplus \diamond_{i_{j+1}} \oplus\text{-SAT}(D_{j+1}) \oplus \dots \oplus \diamond_{i_n} \oplus\text{-SAT}(D_n) \oplus (\oplus\text{-SAT}(\varphi_2)), \end{aligned}$$

we can replace E with 0 in C_1 and C_2 , and get a counter-example with less diamonds than $l_1 + l_2$, which is a contradiction. Therefore, all of the D_j are pairwise \mathcal{F} -inequivalent and satisfiable. $C_1 \oplus C_2 = \diamond_{i_1} D_1 \oplus \dots \oplus \diamond_{i_n} D_n \oplus \varphi_1 \oplus \varphi_2$ is not \mathcal{F} -satisfiable, since $C_1 \equiv_{\mathcal{F}} C_2$. Thus, with Lemma 3.22 it follows that there exist $1 \leq i \neq j \leq n$ such that $D_i \equiv_{\mathcal{F}} D_j$. This is a contradiction.

Thus, it follows that $n \leq 1$. First assume $n = 1$. Then $C_1 = \diamond_{i_1} D_1 \oplus \varphi_1 \equiv_{\mathcal{F}} \varphi_2 = C_2$. This is equivalent to $\varphi_1 \equiv_{\mathcal{F}} \varphi_2$ ($\varphi_1 \equiv_{\mathcal{F}} \neg\varphi_2$) and D_1 is not \mathcal{F} -satisfiable (an \mathcal{F} -tautology). Thus, D_1 is not \mathcal{F} -satisfiable (an \mathcal{F} -tautology), which is a contradiction to the above.

Therefore $n = 0$, and both circuits are propositional (since any occurring \diamond -gates that are not connected to the output-gate are removed by the algorithm), and $\oplus\text{-SAT}(C_i) = C_i$. In this case, $\oplus\text{-SAT}$ rewrites the input circuits as formulas, orders the appearing variables and constants, and deletes double occurrences. The result is a unique formula representation of the input circuit. Thus, the claim holds for $l = 0$, and hence the theorem is proven. \square

It is interesting to note that since the algorithm never adds a gate to a circuit, Theorem 3.20 implies that for a given input circuit, the algorithm computes a smallest possible circuit representing the same function. This implies that minimization problems in this context can be decided in polynomial time as well. Note that the algorithm, when given a formula as input, also returns a formula. Therefore, it can be used to minimize both circuits and formulas.

The above proof does not generalize to other classes of frames. The main reason is that no analog of Lemma 3.22 seems to hold for classes of frames involving, for example, reflexivity or transitivity. While we conjecture that the corresponding problem for these classes of frames can still be solved in polynomial time, we mention that there are examples in the literature that behave differently—sometimes, restricting the class of frames increases the complexity of the modal satisfiability problem. For example, Halpern showed that when considering only formulas of bounded modal nesting degree, the complexity of the satisfiability problem for K drops from PSPACE-complete to NP-complete. On the other hand, for the logic $S4$, the problem remains PSPACE-complete [Hal95]. Therefore syntactical restrictions that reduce the complexity of the general logic K do not necessarily also reduce the complexity for logics defined over a restricted class of models.

Our results for linear propositional functions conclude our discussion about the modal satisfiability problem for the class of frames K in the case that we allow both modal operators in our formulas and circuits: Figure 1 shows that we have covered all clones, and hence reached a complete classification of this problem.

3.5 Satisfiability With Only One Operator

We now look at satisfiability problems with only one of type of operators \Diamond or \Box present. For sets B such that $[B] \supseteq S_1$, we already established PSPACE-completeness for the classes of frames we consider (Corollary 3.11). Since polynomial-time results for the case where we allow both \Box and \Diamond obviously carry over to the case where only one of them is allowed, the following theorem completes a full classification of the problem.

Theorem 3.23. *Let B be a finite set of Boolean functions such that $B \subseteq M$, let $k \geq 0$, and let $M = \{\Box\}$ or $M = \{\Diamond\}$, and let $\mathcal{F} \in \{K, K4\}$. Then $K\text{-CSAT}_M^k(B) \in P$.*

Proof. Due to Lemma 2.6, we can assume that $B = \{\wedge, \vee, 0, 1\}$. We now show that in the case $M = \{\Diamond\}$, a circuit $C \in \text{MCIRC}_\Diamond^k(B)$ is \mathcal{F} -satisfiable if and only if it is satisfied in the reflexive singleton where each variable is set to true, and in the case $M = \{\Box\}$, every $C \in \text{MCIRC}_\Box^k(B)$ is \mathcal{F} -satisfiable if and only if it is satisfied in the irreflexive singleton with every variable set to true (since both the reflexive and the irreflexive singleton are \mathcal{F} -models, the “if” direction of this claim is trivial). These conditions obviously can be tested in polynomial time.

We show the claim by induction on the structure of the formula expansion of the circuit. If C is a single variable or a constant, then the claim obviously holds. Now assume that $C = C_1 \vee C_2$. If C is satisfiable, then at least one of C_1, C_2 is satisfiable, and due to induction, they are satisfied in the reflexive resp. irreflexive singleton with every variable set to true. If $C = C_1 \wedge C_2$, and C is satisfiable then both C_1 and C_2 are satisfiable. By induction, both of them are satisfied in the singleton with every variable set to true. Hence, C is satisfied in this singleton as well.

For the modal operators, assume that $C = \Diamond_i D$ for some $i \in \{1, \dots, k\}$. If C is satisfiable, then obviously D is satisfiable as well, and by induction, D is satisfiable in the reflexive singleton with every variable set to true. For this case, C obviously is satisfied in the same model.

Finally, if $C = \Box_i D$ for some $i \in \{1, \dots, k\}$, then by definition C is satisfied in the irreflexive singleton with every variable set to true. \square

4 The Validity Problem

Besides the satisfiability problem, another problem which often is of interest is the validity problem, i.e., the problem to decide whether a given formula is valid, or is a tautology in a given logic. Recall that in our context, a formula φ is an \mathcal{F} -tautology if and only if φ is \mathcal{F} -equivalent to 1 (this is the case if and only if φ holds in every world of every \mathcal{F} -model).

It is obvious that a formula φ is a tautology if and only if $\neg\varphi$ is not satisfiable. With this easy observation, the complexity of the satisfiability problem and that of the validity problem often can be related to each other—they are “duals” of each other. However, in the case of restricted propositional bases, we cannot always express negation, which is necessary in order to do the transformation mentioned above directly. Therefore, we consider a more general notion of duality, which is closely related to the self-dual property defined for functions earlier: A function f is self-dual if and only if $\text{dual}(f) = f$.

Definition 4.1. *Let f be an n -ary Boolean function. Then $\text{dual}(f)$ is the n -ary Boolean function defined as $\text{dual}(f)(x_1, \dots, x_n) = \neg f(\overline{x_1}, \dots, \overline{x_n})$.*

For a set B of Boolean functions, $\text{dual}(B)$ is defined as the set $\{\text{dual}(f) \mid f \in B\}$. Obviously, a similar duality exists between the modal operators \Diamond and \Box : For a set $M \subseteq \{\Box, \Diamond\}$, we define $\text{dual}(M)$ to be the set such that $\Box \in \text{dual}(M)$ if and only if $\Diamond \in M$, and $\Diamond \in \text{dual}(M)$ if and only if $\Box \in M$. For a clone B , the dual clone $\text{dual}(B)$ can easily be identified in Post’s Lattice (see Figure 1), as it is simply the “mirror class” with regard to the vertical symmetry axis in the lattice. The following theorem shows that complexity classifications for the satisfiability problem immediately give dual classifications for the validity problem.

Theorem 4.2. *Let B be a finite set of Boolean functions, let $k \geq 0$, and let \mathcal{F} be a class of frames, and let $M \subseteq \{\Box, \Diamond\}$. Then the following holds:*

1. $\mathcal{F}\text{-CTAUT}_M^k(B) \equiv_m^{\log} \overline{\mathcal{F}\text{-CSAT}_{\text{dual}(M)}^k(\text{dual}(B))}$.
2. $\mathcal{F}\text{-FTAUT}_M^k(B) \equiv_m^{\log} \overline{\mathcal{F}\text{-FSAT}_{\text{dual}(M)}^k(\text{dual}(B))}$.

Proof. Let C be a circuit from $\text{MCIRC}_M^k(B)$. We construct the circuit $\text{dual}(C)$ by exchanging every f -gate for a function $f \in B$ with a $\text{dual}(f)$ -gate. Similarly, we replace every \Box_i -gate with a \Diamond_i -gate, and vice versa. It is obvious that this transformation can be performed in logarithmic space, and that the same transformation can be applied to formulas.

It remains to prove that C is unsatisfiable if and only if $\text{dual}(C)$ is a tautology. Since $\text{dual}(\cdot)$ is obviously injective, and $\text{dual}(\text{dual}(C)) = C$, this also proves that C is a tautology if and only if $\text{dual}(C)$ is unsatisfiable, and hence proves the reduction.

Inductively, we show a more general statement: For any modal model M , let $\neg M$ denote the model obtained from M by reversing the propositional truth assignment, i.e., where a variable in a world is true if and only if the same variable is false in the same world in M . We show that for any model M and any world $w \in M$, it holds that $M, w \models C$ if and only if $\neg M, w \not\models \text{dual}(C)$. This obviously completes the proof, since M is an \mathcal{F} -model if and only if $\neg M$ is.

We show the claim by induction on the structure of C . First, assume that C is equivalent to the variable x_i . Then $M, w \models C$ if and only if $M, w \models x_i$ if and only if $\neg M, w \not\models x_i$. Since $\text{dual}(x_i) = x_i$, this proves the base step.

Now assume that the output gate g of C is an f -gate for an n -ary Boolean function $f \in B$, and let g_1, \dots, g_n be the predecessor gates of g in C . By induction, we know that for each $j \in \{1, \dots, n\}$, it holds that $M, w \models C_{g_j}$ if and only if $\neg M, w \not\models \text{dual}(C_{g_j})$ (where C_{g_j} is the subcircuit of C with output gate g_j). For $j \in \{1, \dots, n\}$, let α_j be defined as 1 if $M, w \models C_{g_j}$, and 0 otherwise. By induction, we know that α_j is 1 if and only if $\neg M, w \not\models \text{dual}(C)$. Now observe that $M, w \models C$ if and only if $f(\alpha_1, \dots, \alpha_n) = 1$, if and only if $\text{dual}(f)(\overline{\alpha_1}, \dots, \overline{\alpha_n}) = 0$, and this is the case if and only if $\neg M, w \not\models \text{dual}(C)$.

Now assume that the output gate g of C is a \Diamond_i -gate for some $i \in \{1, \dots, k\}$, and let h be the predecessor gate of g in C . Then the following holds:

$$\begin{aligned}
M, w \models C &\text{ iff there is a world } w' \text{ such that } (w, w') \in R_i \text{ and } M, w' \models C_h \\
&\text{ iff there is a world } w' \text{ such that } (w, w') \in R_i \text{ and } \neg M, w' \not\models \text{dual}(C_h) \\
&\text{ iff } \neg M, w \not\models \Box_i \text{dual}(C_h) \\
&\text{ iff } \neg M, w \not\models \text{dual}(C).
\end{aligned}$$

Finally, assume that the output gate g of C is a \Box_i -gate for some $i \in \{1, \dots, k\}$, and let h be the predecessor gate of g in C . Then the following holds:

$$\begin{aligned}
M, w \models C &\text{ iff for each world } w' \text{ such that } (w, w') \in R_i, M, w' \models C_h \\
&\text{ iff for each world } w' \text{ such that } (w, w') \in R_i, \neg M, w' \not\models \text{dual}(C_h) \\
&\text{ iff } \neg M, w \not\models \Diamond_i \text{dual}(C_h) \\
&\text{ iff } \neg M, w \not\models \text{dual}(C).
\end{aligned}$$

This concludes the induction, and therefore the proof. \square

5 Conclusion and Further Research

We completely classified the complexity of the modal satisfiability and validity problems arising when restricting the allowed propositional operators in the formula for the logics K and KD. We showed that the more succinct representation of modal formulas as circuits does not have an effect on the complexity of these problems up to \leq_m^p -degree. We also showed that for multi-modal logics, the results only depend on whether we have 0, 1, or 2 modalities, adding more modal operators does not increase the complexity of the problems we studied. Note that in many cases, our results hold for more general classes of frames, as often, they are stated for any class containing the reflexive singleton, or similar conditions. This does not only apply to most of our polynomial-time results, but also for our circuit-to-formula construction in Corollary 3.7, and our implementation results in Theorem 3.10 and the uni-modal version of Theorem 3.9.

The most obvious next question to look at is to complete our complexity classification for other classes of frames. For $\mathcal{F} \in \{T, S4, S5\}$, our proofs already give a complete classification with the exception of the complexity of the problems $\mathcal{F}\text{-FSAT}_M^k(B)$ and $\mathcal{F}\text{-CSAT}_M^k(B)$ where $[B] \in \{L_0, L_1\}$. We conjecture that these cases are solvable in polynomial time as well, however, to solve these cases different ideas from the ones used in the proof for K and KD are required. Another interesting question is the exact complexity of our polynomial cases, most notably the case where the propositional operators represent linear functions.

There are many other interesting directions for future research. For example, one can look at other decision problems (e.g., global satisfiability and formula minimization), and one can try to generalize modal logic modally as well as propositionally.

6 Acknowledgments

We thank Michael Bauland for his work on the work presented in [BHSS06], and Thomas Schneider and Heribert Vollmer for helpful discussions. We also thank the anonymous STACS referees for their helpful comments and suggestions, and Steffen Reith for providing the figure of Post’s Lattice.

References

- [BCRV03] E. Böhler, N. Creignou, S. Reith, and H. Vollmer. Playing with Boolean blocks, part I: Post’s lattice with applications to complexity theory. *SIGACT News*, 34(4):38–52, 2003.
- [BCRV04] E. Böhler, N. Creignou, S. Reith, and H. Vollmer. Playing with Boolean blocks, part II: Constraint satisfaction problems. *SIGACT News*, 35(1):22–35, 2004.
- [BdRV01] P. Blackburn, M. de Rijke, and Y. Venema. *Modal logic*. Cambridge University Press, New York, NY, USA, 2001.
- [BG04] B. Bennett and A. Galton. A unifying semantics for time and events. *Artificial Intelligence*, 153(1-2):13–48, 2004.

- [BHSS06] M. Bauland, E. Hemaspaandra, H. Schnoor, and I. Schnoor. Generalized modal satisfiability. In *Proceedings of the 23rd Symposium on Theoretical Aspects of Computer Science, LNCS*, pages 500–511, 2006.
- [BMS⁺07] M. Bauland, M. Mundhenk, T. Schneider, H. Schnoor, I. Schnoor, and H. Vollmer. The tractability of model-checking for LTL: The good, the bad, and the ugly fragments. In *Proceedings of Methods for Modalities 5, 2007, ENTCS*, 2007. to appear.
- [BSS⁺07] M. Bauland, T. Schneider, H. Schnoor, I. Schnoor, and H. Vollmer. The complexity of generalized satisfiability for linear temporal logic. In *Foundations of Software Science and Computational Structures, LNCS*, pages 48–62. Springer, 2007.
- [CDF03] T. Coffey, R. Dojen, and T. Flanagan. On the automated implementation of modal logics used to verify security protocols. In *ISICT '03: Proceedings of the 1st international symposium on Information and communication technologies*, pages 329–334. Trinity College Dublin, 2003.
- [Dal00] V. Dalmau. *Computational Complexity of Problems over Generalized Formulas*. PhD thesis, Departament de Llenguatges i Sistemes Informàtica, Universitat Politècnica de Catalunya, 2000.
- [DHL⁺92] F. Donini, B. Hollunder, M. Lenzerini, D. Nardi, W. Nutt, and A. Spaccamela. The complexity of existential quantification in concept languages. *Artificial Intelligence*, 53(2-3):309–327, 1992.
- [DLNN97] F. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. *Information and Computation*, 134:1–58, 1997.
- [FHJ02] U. Frendrup, Hüttel, and J. Jensen. Modal logics for cryptographic processes. In *Proceedings of EXPRESS 02*, 2002.
- [Hal95] J. Halpern. The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic. *Artificial Intelligence*, 75(2):361–372, 1995.
- [Hem96] E. Hemaspaandra. The price of universality. *Notre Dame Journal of Formal Logic*, 37(2):174–203, 1996.
- [Hem01] E. Hemaspaandra. The complexity of poor man’s logic. *Journal of Logic and Computation*, 11(4):609–622, 2001. Corrected version: [Hem05].
- [Hem05] E. Hemaspaandra. The Complexity of Poor Man’s Logic, *CoRR*, cs.LO/9911014, 1999. Revised 2005.
- [HM92] J. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(2):319–379, 1992.
- [HMT88] J. Halpern, Y. Moses, and M. Tuttle. A knowledge-based analysis of zero knowledge. In *STOC '88: Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 132–147, New York, NY, USA, 1988. ACM Press.
- [HS08] E. Hemaspaandra and H. Schnoor. On the complexity of elementary modal logics. In *Proceedings of the 25th Symposium on Theoretical Aspects of Computer Science*, volume 08001 of *Dagstuhl Seminar Proceedings*, pages 349–360. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2008.
- [JCG97] P. Jeavons, D. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44(4):527–548, 1997.
- [Lad77] R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6(3):467–480, 1977.

- [Lau06] D. Lau. *Function Algebras on Finite Sets: Basic Course on Many-Valued Logic and Clone Theory (Springer Monographs in Mathematics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [Lew79] H. Lewis. Satisfiability problems for propositional calculi. *Mathematical Systems Theory*, 13:45–53, 1979.
- [Lia03] C.-J. Liao. Belief, information acquisition, and trust in multi-agent systems – a modal logic formulation. *Artificial Intelligence*, 149(1):31–60, 2003.
- [LR86] R. Ladner and J. Reif. The logic of distributed protocols: Preliminary report. In *TARK '86: Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 207–222, San Francisco, CA, USA, 1986. Morgan Kaufmann Publishers Inc.
- [Moo79] R. Moore. Reasoning about knowledge and action. Technical Report 191, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, 1979.
- [MSHI78] J. McCarthy, M. Sato, T. Hayashi, and S. Igarashi. On the model theory of knowledge. Technical report, Stanford, CA, USA, 1978.
- [Pos41] E. Post. The two-valued iterative systems of mathematical logic. *Annals of Mathematical Studies*, 5:1–122, 1941.
- [Rei01] S. Reith. *Generalized Satisfiability Problems*. PhD thesis, Fachbereich Mathematik und Informatik, Universität Würzburg, 2001.
- [RV00] S. Reith and H. Vollmer. Optimal satisfiability for propositional calculi and constraint satisfaction problems. In *Proceedings of the 25th International Symposium on Mathematical Foundations of Computer Science, LNCS*, pages 640–649. Springer Verlag, 2000.
- [RW05] S. Reith and K. Wagner. The complexity of problems defined by Boolean circuits. In *Proceedings of Mathematical Foundations of Informatics 1999*. World Science Publishing, 2005.
- [SC85] A. Sistla and E. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, 1985.
- [Sch78] T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings 10th Symposium on Theory of Computing*, pages 216–226. ACM Press, 1978.
- [Sch07] H. Schnoor. *Algebraic Techniques for Satisfiability Problems*. PhD thesis, University of Hannover, 2007.
- [SP06] L. Schröder and D. Pattinson. PSPACE bounds for rank-1 modal logics. In *LICS*, pages 231–242, 2006.
- [SS91] M. Schmidt-Schauss and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.
- [Vol99] H. Vollmer. *Introduction to Circuit Complexity – A Uniform Approach*. Texts in Theoretical Computer Science. Springer Verlag, Berlin Heidelberg, 1999.